

What is claimed is:

CLAIMS

1. A method of processing native events by a virtual machine that operates on a first platform, wherein said virtual machine concurrently supports a first and a second task, said method comprising:

receiving, by a virtual machine, a native event that is associated with a first platform;

selecting, by said virtual machine, one of said first and second tasks as a selected task for receiving said native event, wherein said selection is performed when said first and second tasks are concurrently supported by said virtual machine; and

processing said native event on behalf of said task.

2. A method as recited in claim 1, wherein said method further comprises:

providing an event-repository and an event-handler for said selected task; and

placing said native event in said event-repository;

invoking said event-handler to initiate processing of said native event; and

processing, by said event-handler, said native event.

3. A method as recited in claim 2,

wherein said event-repository is implemented as a first-in first-out queue,

wherein said event-handler is implemented as an event-handler thread, and

wherein said selection is performed by an event-manager thread.

4. A method as recited as claim 1, wherein said method further comprises:
manipulating said native event to be Java compliant.
5. A method as recited in claim 1, wherein said manipulating of said native event is performed by said virtual machine.
6. A method as recited in claim 1, wherein said selected task is a Java compliant MIDlet.
7. A method as recited in claim 1, wherein said selected task is a portion of a Java compliant MIDlet.
8. A method as recited in claim 1, wherein said selection comprises:
selecting a foreground task when said selection is made.
9. A method as recited in claim 1,
wherein said selected task is a Java compliant MIDlet, and
wherein said selection comprises:
selecting a foreground task when said selection is made.
10. A method as recited in claim 9, wherein said selecting said foreground tasks comprises: selecting a task that is displayed for the user.
11. A method as recited in claim 10, wherein said first platform includes a mobile device.

12. A virtual machine for processing native events associated with a first platform, wherein said virtual machine concurrently supports a first and a second task, said virtual machine comprising:

first and second tasks that are concurrently operating on said virtual machine; and

a native event dispatcher that can operate to:

receive a native event that is associated with said first platform; and

select one of said first and second tasks as a selected task for processing said native event.

13. A virtual machine as recited in claim 12, wherein said virtual machine further comprises: a native event-handler for said selected task, wherein said native event-handler is invoked when said selected task is selected, and wherein said native event-handler processes said native event when said native event-handler is invoked.

14. A virtual machine as recited in claim 12, wherein said virtual machine further comprises:

a native event-repository and a native event-handler for said selected task; and

said native event-dispatcher can further operate to:

place said native event in said native event-repository; and

invoke said native event-handler to initiate processing of said native event.

15. A method of processing native events by a virtual machine that operates on a first platform, wherein said virtual machine concurrently

supports a first and a second task on said first platform, said method comprising:

receiving, by a virtual machine, a native event that is associated with a first platform;

determining, by said virtual machine, which one of said first and second tasks is a foreground task, wherein said foreground task is the only task that is displayed; and

processing, by said foreground task, said native event.

16. A method as recited in claim 15, wherein said first platform includes a mobile device.

17. A method as recited in claim 15, wherein said first and second tasks are MIDlets.

18. A virtual machine capable of processing native events, wherein said virtual machine concurrently supports a first and a second task on a first platform, and wherein said virtual machine can operate to:

receive a native event that is associated with said first platform;

determine which one of said first and second tasks is the task displayed;

process said foreground task on behalf of said native event.

19. A virtual machine as recited in claim 18,
wherein said virtual machine operates on a mobile device; and
wherein said foreground task is a Java complaint MIDlet.

20. A computer readable medium including computer program for processing native events by a virtual machine that operates on a first platform, wherein said virtual machine concurrently supports a first and a second task, said computer readable medium comprising:

computer program code for receiving a native event that is associated with a first platform;

computer program code for selecting one of said first and second tasks as a selected task for receiving said native event, wherein said selection is performed when said first and second tasks are concurrently supported by said virtual machine; and

computer program code for processing said native event.

21. A computer readable medium as recited in claim 20, wherein said readable medium further comprises:

computer program code for manipulating said native event, wherein said manipulating comprises: representing said native event in a form that is accessible by said selected task.

22. A computer readable medium as recited in claim 20, wherein said selected task is a Java compliant MIDlet.